CLAIMS

What is claimed is:

1. A method of executing a sequence of instructions comprising:

comparing a first plurality of least significant bits (LSBs) of a first operand to a second plurality of LSBs of a second operand to determine a predicted predicate value (PPV) for a predicate;

and

conditionally executing a predicated instruction depending on the PPV.

2. The method of claim 1, further comprising:

executing a COMPARE instruction to determine an actual predicate value (APV) for the predicate;

comparing the APV to the PPV; and

flushing a pipeline if the APV and the PPV are unequal.

3. The method of claim 2, further comprising executing the predicated instruction after flushing the pipeline.

4. The method of claim 2, wherein flushing the pipeline comprises flushing only a backend portion of the pipeline.

5. The method of claim 2, further comprising updating historical information corresponding to the predicate in a predicate history table after comparing the APV to the PPV.

6. The method of claim 1, wherein comparing the first plurality to the second plurality includes comparing the least significant quarter or less of the first operand bits to the least significant quarter or less of the second operand bits.

7. The method of claim 1, wherein conditionally executing the predicated instruction includes executing the predicated instruction if the PPV is true.

8. The method of claim 7, wherein conditionally executing the predicated instruction includes treating the predicated instruction like a no-op if the PPV is false.

9. The method of claim 1, wherein comparing the first plurality to the second plurality is done after determining a low confidence level in an ability to accurately calculate the PPV based on historical information associated with the predicate.

10. A method of executing a sequence of instructions comprising:

determining a confidence level in an ability to accurately calculate a predicted predicate value (PPV) for a predicate based on historical information associated with the predicate;

determining the PPV using the historical information if the confidence level is determined to be a first level; and

determining the PPV by comparing a first plurality of least significant bits (LSBs) of a first operand of a COMPARE instruction to a second plurality of LSBs of a second operand of the COMPARE instruction if the confidence level is determined to be a second level.

11. The method of claim 10, further comprising:

conditionally executing a predicated instruction depending on the PPV;

determining an actual predicate value (APV) for the predicate; and

flushing a backend of a pipeline if the APV and the PPV are unequal.

12. The method of claim 10, wherein determining the PPV by comparing includes comparing the lowest eighth or less of LSBs of the first operand to the lowest eighth or less of LSBs of the second operand.

13. The method of claim 10, wherein determining the PPV by comparing includes comparing for equality between the first and second pluralities.

14. A processor comprising:

a predicate history table;

a register file;

a predicted predicate value (PPV) calculator having a first input coupled to an output of the predicate history table and a second input coupled to an output of the register file; and

a speculative predicate register file coupled to an output of the calculator.

15. The processor of claim 14, further comprising:

a IP select circuit having an output coupled to the predicate history table;

a register select circuit having an output coupled to the register file; and

an instruction decoder having an output coupled to input of the IP select circuit and the register select circuit.

16. The processor of claim 15, further comprising a pipeline having a PPV input coupled to an output of the file and an actual predicate value (APV) output coupled to an input of the predicate history table.

17. The processor of claim 16, further comprising an XOR gate having a first input coupled to the APV output of the pipeline, a second input coupled to an output of the file, and an output coupled to a flush input of the pipeline.

18. A processor comprising:

a predicate history table to store historical information associated with a predicate; and

a predicted predicate value (PPV) calculator to calculate a PPV based on a comparison of a first plurality of least significant bits (LSBs) of a first operand to a second plurality of LSBs of a second operand.

19. The processor of claim 18, further comprising a speculative predicate register file to store the PPV.

20. The processor of claim 19, further comprising a pipeline to receive the PPV, and to conditionally execute a predicated instruction depending on the PPV.

21. The processor of claim 18, further comprising a pipeline to receive the PPV, and to conditionally execute a predicated instruction depending on the PPV.

22. The processor of claim 21, wherein the pipeline includes an actual predicate value output to provide an actual predicate value to the predicate history table.

23. The processor of claim 18, wherein the calculator includes a selector to select, based on a confidence level, the PPV to be based on either historical information or the comparison.

24. A system comprising:

memory to store a first value;

a bus to transfer the first value from the memory; and

a processor to receive the first value from the bus and to compare a first plurality of least significant bits (LSBs) of the first value to a second plurality of least significant bits (LSBs) of a second value to calculate a predicted predicate value (PPV) for a predicate.

25. The system of claim 24, wherein the memory is main memory.

26. The system of claim 24, wherein the bus is a system bus.

27. The system of claim 24, wherein the processor further comprises a pipeline to receive the PPV, and to conditionally execute a predicated instruction depending on the PPV.

28. The system of claim 25, wherein the processor further comprises a predicate history table to store historical information associated with the predicate.

ADD A2